WHAT IS CLAIMED IS:

1.   A method for transmitting local node function parameters to a remote node for execution of the function on the remote node, comprising:

associating a representation string with function parameters on a first stack, wherein each character in the representation string corresponds to the data type of an individual function parameter on the first stack;

dereferencing pointer parameters on the first stack;

generating a pure value buffer with the function parameters and the dereferenced pointers;

transmitting the pure value buffer to the remote node.

2.   The method of claim 1, wherein associating the representation string further comprises generating a DTSTRUCT string.

3.   The method of claim 1, wherein associating the representation string further comprises assigning a specific text string character to each function parameter data type on the first stack, wherein the assigning is conducted by at least one of a user input and a compiler generation operation.

4.   The method of claim 1, wherein dereferencing pointer parameters on the first stack further comprises retrieving data represented by the pointer parameters and placing the data represented by the pointer parameters on the pure value buffer.

5.   The method of claim 1, further comprising optimizing the pure value buffer prior to transmitting to the remote node.

6.   The method of claim 5, wherein optimizing further comprises eliminating remote node write only-type data from the pure value buffer prior to transmitting the pure value buffer to the remote node.

7.   The method of claim 5, wherein optimizing further comprises eliminating local node read only-type data from the pure value buffer prior to transmitting the pure value

buffer to the remote node.

8.   The method of claim 1, further comprising:

receiving the pure value buffer at the remote node;

generating a second stack on the second node mirroring the first stack on the first node;

executing a function using the second stack;

creating a return pure value buffer; and

transmitting the return pure value buffer to the first node.

9.   The method of claim 8, wherein generating the second stack further comprises using the representation string to recreate the second stack from the pure value buffer.

10.   The method of claim 8, further comprising:

receiving return pure value buffer on the first node;

regenerating the first stack on the first node; and

replacing each pointer that was written back in an original memory location pointed to by the first stack.

11.   A method for transmitting function related data from a first node to a second node, comprising:

generating a stack having the function related data stored thereon;

dereferencing pointer parameters in the function related data with a representation structure;

generating a pure value buffer; and

transmitting the pure value buffer to the second node.

12.   The method of claim 11, wherein generating a stack having the function related data further comprises retrieving the function related data from various memory locations and storing the function related data in a contiguous stack location.

13.   The method of claim 11, wherein dereferencing pointer parameters further

comprises retrieving pure value data represented by the pointer parameters and storing the pure value data on the stack.

14.   The method of claim 11, wherein generating a pure value buffer further comprises inserting pure value data into the stack in place of the pointer parameters and copying the stack contents to the pure value buffer.

15.   The method of claim 11, further comprising optimizing the pure value buffer prior to transmitting to the second node.

16.   The method of claim 12, wherein optimizing further comprises eliminating second node write only-type data from the pure value buffer prior to transmitting the pure value buffer to the second node.

17.   The method of claim 12, wherein optimizing further comprises eliminating first node read only-type data from the pure value buffer prior to transmitting the pure value buffer to the second node.

18.   A computer readable medium storing a software program that, when executed by a processor, causes the processor to perform a method comprising:

associating a representation string with function parameters on a first stack, wherein each character in the representation string corresponds to the data type of an individual function parameter on the first stack;

dereferencing pointer parameters on the first stack;

generating a pure value buffer with the function parameters and the dereferenced pointers;

transmitting the pure value buffer to the remote node.

19.   The computer readable medium of claim 18, wherein associating the representation string further comprises generating a DTSTRUCT string.

20.   The computer readable medium of claim 18, wherein associating the

33

representation string further comprises assigning a specific text string character to each function parameter data type on the first stack, wherein the assigning is conducted by at least one of a user input and a compiler generation operation.

21.   The computer readable medium of claim 18 wherein dereferencing pointer parameters on the first stack further comprises retrieving data represented by the pointer parameters and placing the data represented by the pointer parameters on the pure value buffer.

22.   The computer readable medium of claim 18, further comprising optimizing the pure value buffer prior to transmitting to the remote node.

23.   The computer readable medium of claim 22, wherein optimizing further comprises eliminating remote node write only-type data from the pure value buffer prior to transmitting the pure value buffer to the remote node.

24.   The computer readable medium of claim 22, wherein optimizing further comprises eliminating local node read only-type data from the pure value buffer prior to transmitting the pure value buffer to the remote node.

25.   The computer readable medium of claim 18, further comprising:
            receiving the pure value buffer at the remote node;
            generating a second stack on the second node mirroring the first stack on the first node;
            executing a function using the second stack;
            creating a return pure value buffer; and
            transmitting the return pure value buffer to the first node.

26.   The computer readable medium of claim 25, wherein generating the second stack further comprises using the representation string to recreate the second stack from the pure value buffer.

27.   The computer readable medium of claim 25, further comprising:

receiving return pure value buffer on the first node;

regenerating the first stack on the first node; and

replacing each pointer that was written back in an original memory location pointed to by the first stack.

28.   A computer readable medium storing a software program that, when executed by a processor, causes the processor to perform a method comprising:

generating a stack having function related data stored thereon;

dereferencing pointer parameters in the function related data with a representation string;

generating a pure value buffer; and

transmitting the pure value buffer to the second node.

29.   The computer readable medium of claim 28, wherein generating a stack having the function related data further comprises retrieving the function related data from various memory locations and storing the function related data in a contiguous stack location.

30.   The computer readable medium of claim 28, wherein dereferencing pointer parameters further comprises retrieving pure value data represented by the pointer parameters and storing the pure value data on the stack.

31.   The computer readable medium of claim 28, wherein generating a pure value buffer further comprises inserting pure value data into the stack in place of the pointer parameters and copying the stack contents to the pure value buffer.

32.   The computer readable medium of claim 28, further comprising optimizing the pure value buffer prior to transmitting to the second node.

33.   The computer readable medium of claim 29, wherein optimizing further comprises eliminating second node write only-type data from the pure value buffer prior to

transmitting the pure value buffer to the second node.

34.   The computer readable medium of claim 29, wherein optimizing further comprises eliminating first node read only-type data from the pure value buffer prior to transmitting the pure value buffer to the second node.